Classwise Domain Generalization: A Novel Framework for Evaluating OOD Performance

*Sarath Sivaprasad[1] *Akshay Goindani[2] Mario Fritz[1] Vineet Gandhi[2]
[1]CISPA Helmholtz Center for Information Security [2]CVIT, IIIT Hyderabad
{sarath.s, fritz}.@cispa.de akshay.goindani@alumni@iiit.ac.in vgandhi@iiit.ac.in

Abstract

Given that neural networks generalize unreasonably well in the IID setting, Out-Of-Distribution(OOD) evaluation presents a useful failure case to study their generalization performance. Recent studies have shown that a carefully trained ERM gives good performance in Domain Generalization (DG) Gulrajani & Lopez-Paz (2021), with train samples from all domains randomly shuffled in each batch of training. Furthermore, Later studies have shown DG specific methods to boost the test performance of neural networks under distribution shift without training data being explicitly annotated with domain information. This observation is counter-intuitive as the studies on the failure cases of OOD has shown that, without being trained with domain knowledge, neural networks will fit domain specific features for reducing train loss. We present a new setting beyond the Traditional DG (TDG) called the Class-wise DG (CWDG), where for each class, we randomly select one of the domains and keep it aside for testing. Despite being exposed to all domains during training, our experiments show that the performance of the neural network drops in this framework compared to TDG. We evaluate popular DG methods and show that the performance of different methods under TDG and CWDG setting are not correlated. Finally, we propose a novel method called Iterative Domain Feature Masking (IDFM) which uses domain annotations in the train data, achieving state-of-the-art results on the proposed benchmark.

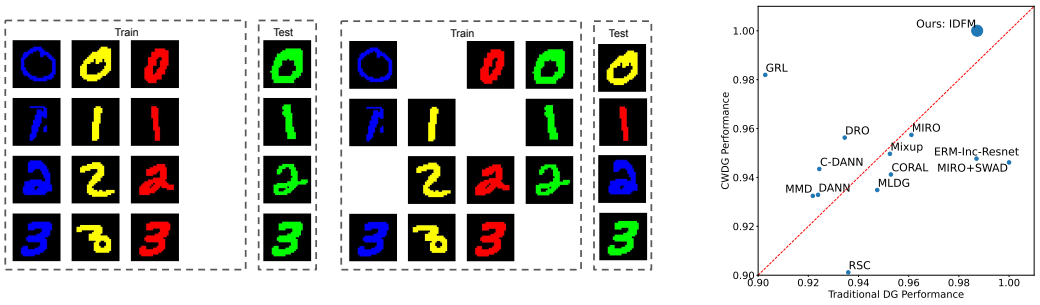
1 Introduction

Many real-world applications require neural networks to be robust to distribution shifts in test data. These shifts are often unavoidable in the wild, but Neural Networks have shown to substantially drop performance in these scenarios. Domain Generalization (DG) is a setting designed to evaluate the robustness of a model for such challenges. DG formulation follows training on data sampled from the same set of classes from all available domains in the dataset except one domain, which is kept out for testing. The goal here is to learn a model from the multiple related domains and classify the same classes in an unseen domain. For instance, the model is trained with sets of photos, paintings, and cartoons and evaluated on sketches Li et al. (2017).

Traditional DG (TDG) is formulated as an optimization problem where the parameters are tuned to minimize the expected loss of classification over the set of all possible domains in which the given classes can be meaningfully represented. This is a more challenging optimization problem compared to Domain Adaptation and other formulations to evaluate distribution shifts, where there are some assumptions on the target distribution. Moreover, TDG makes a strong assumption that data is available in all classes for all the domains. This assumption is reflected in the statistics of datasets in benchmarks like Domainbed Gulrajani & Lopez-Paz (2021). This assumption is often difficult to meet in many real-world scenarios.

*These authors contributed equally to this work

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Traditional DG (b) Class-wise DG (c) TDG vs CWDG

Figure 1: The figure is an illustration of the difference in train test split in the Traditional DG (TDG) setting and the proposed Class-wise DG (CWDG) setting . The color of the MNIST digits depicts the domain and the dotted squares show the train test splits. (a) shows one of the four splits in TDG, and (b) shows one train-test split of CWDG out of the possible samplings. The open entry in the train set of CWDG corresponds to the entry in the test set, and (c) shows the normalized performance of various methods on the TDG and CWDG evaluations averaged over six popular DG datasets.

In real-world applications, we often cannot have all classes annotated in all the domains. We propose a framework for evaluating the robustness of models under such skewed domain-class joint distributions. For each class, we randomly select one of the domains and keep it aside for testing (Figure 1(b)). We call this setting Class-wise Domain Generalization (CWDG). For each class, the challenge is the same as traditional DG. Despite being exposed to all domains during training, our experiments show that the proposed method is more challenging than the TDG evaluation.

Over the years, various inventive methods have been proposed for TDG. Wang et al. Wang et al. (2021) present a comprehensive review of over 150 methods that improve TDG perfromance on multiple benchmarks. Recently, Gulrajani and Lopez-Paz Gulrajani & Lopez-Paz (2021) have shown that an Empirical Risk Minimization (ERM) baseline gives a competent performance on TDG benchmarks, and none of the tailored methods evaluated give any clear and consistent advantage over the baseline.

The evaluation on CWDG shows that many of the newer methods improve performance in TDG but fail to give a similar improvement in CWDG (Figure 1(c)). Earlier methods like Gradient Reversal Ganin & Lempitsky (2015) work better than most of the new state-of-the-art methods on TDG. The presence of multiple domains incentivizes learning domain agnostic features and is the primary reason for generalization in TDG setting Gulrajani & Lopez-Paz (2021). On the contrary, in the proposed setting, the network has an incentive to learn domain-specific features due to the skewed class-domain distribution. Our study demonstrates how some of the prior art despite improving TDG fails to get competitive performance in CWDG. We propose a method called Iterate Domain Feature Masking (IDFM) and show that it achieves state-of-the-art results in the CWDG framework while retaining the performance of ERM in TDG. Overall, our work makes the following contributions:

- We propose a new evaluation benchmark (CWDG), which is relevant to many real-world applications. It amplifies the skewness in the distribution of classes across domains.
- We formulate the CWDG counterparts for popular DG benchmark datasets and evaluate the different DG methods on them.
- We propose a method that improves neural network performance in the proposed CWDG setting while retaining ERM performance in traditional DG.

2 Related Work

We give a more comprehensive review of the evaluation methods for Independent and Identically Distributed (IID) and Out-Of-Distribution (OOD) test data performance in Appendix A. In this section, we briefly discuss the latest related work.

TDG formulation aims at evaluating a model's ability to learn discriminative features from multiple domains and testing on an unseen domain. TDG on image classification is commonly evaluated on six

datasets: Li et al. (2017); Fang et al. (2013); Venkateswara et al. (2017); Arjovsky et al. (2019); Peng et al. (2019); Ghifary et al. (2015). The details of these datasets are given in Table 1 in Appendix B.

Learning domain agnostic features using the TDG formulation has seen significant interest in recent years. The problem has been approached from many different directions, and the prominent ones are data augmentation Borlino et al. (2021); Zhou et al. (2021b); Xu et al. (2020), gradient manipulation Ganin & Lempitsky (2015); Huang et al. (2020), ensemble learning Mancini et al. (2018), and feature disentanglement Khosla et al. (2012); Piratla et al. (2020). For a comprehensive list, readers can refer to the recent surveys Wang et al. (2021); Zhou et al. (2021a). It is worth noting that several of these ideas Ganin & Lempitsky (2015) have found widespread success beyond the TDG setting.

Gulrajani and Lopez-paz Gulrajani & Lopez-Paz (2021) suggest that inconsistencies in experimental conditions (datasets and training protocols) render fair comparisons difficult. They propose DomainBed, a unifying benchmark for TDG, and empirically show that a carefully implemented ERM outperforms all prior art in terms of average performance. However, Nagarajan et al. (2020) show the failure modes of ERM for DG. They show the cases where the model will fail to generalize, unless trained to incorporate the domain information of the samples. Further explorations on DomainBed (MIRO Cha et al. (2022)) benchmark still shuffles train data from all domains and trains without domain information. This motivates to go beyond the TDG setting to better understand the failure modes of NN under distribution shift.

Consequently, we propose CWDG, a more challenging formulation, which leaves room for shortcut learning Geirhos et al. (2020). Our work interestingly contrasts with Maniyar et al. (2020) which proposes further *constraints* on TDG by introducing unseen classes in the test domain. We instead *relax* the assumptions and expose all domains during training.

3 Method

In this section, we explain the formulation of the proposed CWDG setting. Furthermore, we describe the proposed method IDFM which helps prevent neural networks from learning domain specific features.

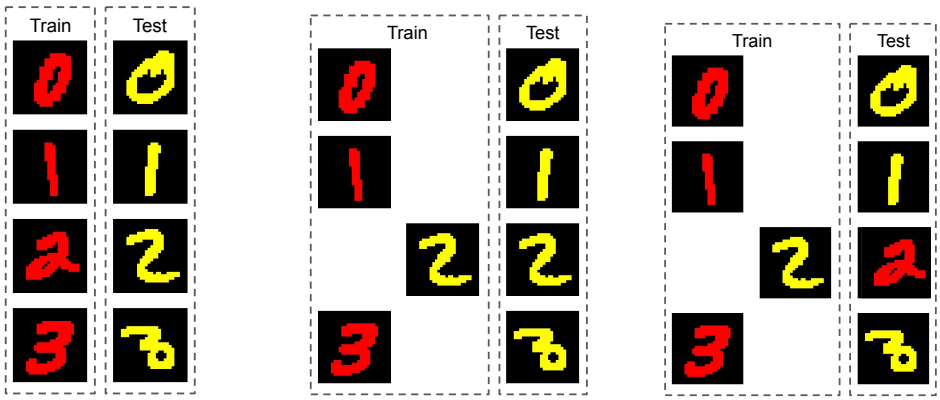
3.1 Proposed Evaluation strategy: CWDG

In the CWDG framework, samples from one domain is kept out of the training set for each of the classes. The model is evaluated on the set of kept out samples. Figure 1(b) is an illustration of the setting where the colors depict the domains and the digits corresponds to the classes. It is to be noted that, this setting does not evaluate the domain generalization capability of the network on the entire dataset, but for each class. That is, the network performance is evaluated on an unseen domain for each class.

Consider the setting shown in Figure 2. Figure 2(a), depicts TDG setting on two domains and four classes. Here the domain shift is uniform across all the classes. Neural networks would gives a performance slightly better than random, depending on the proximity of the domains. In other words the performance improves as the distribution shift reduces.

On the other hand, Figure 1(b) shows a scenario, where samples of a class from the earlier train data is replaced with samples from same class in the test data. Samples of class *2 in domain red from train data is replaced by the samples of class 2 in domain yellow (yellow is the test domain). Despite the distribution shift between train and test data reducing in this setting, it is quite intuitive that all samples in the test set are predicted as the class replaced class (class 2). Here the model can reduce train loss by fitting domain-specific features as there is some information gained about the class from domain features. CWDG is an extension of this framework which creates an incentive for the network to fit domain-specific features.*

Figure 1(c) depicts a CWDG setting of this illustrative example. Once again, it is trivial that all test images from domain yellow is predicted as class 2 and all images from domain red is predicted as classes other than 2. Therefore, the test accuracy of a neural network trained in such setting reduces to zero! Despite being exposed to all domains during training, this illustration show why



(a) Traditional DG (b) Skewed domain-class distribution (c) Towards CWDG

Figure 2: The figure illustrates why CWDG is a more challenging setting than TDG. Here the color depicts the domain and digits show the classes. (a) This is an illustrative example of TDG setting, where domain shift happens uniformly on all classes. (b) Depicts the scenario where, samples from one of the classes (digit 2) in the train data is replaced by samples from a different domain (yellow). (c) Is a possible CWDG seed. That is, in each class one of the domain is kept out from train data towards test set.

the performance of neural network drop. This shows that the challenge of distribution shift can go beyond TDG.

CWDG is a setting where a model has to explicitly learn domain agnostic features, despite the train loss incentivising the same in case of an ERM. It is a more challenging setting for a neural network. This claim is further validated by our experiments. Also, given such shifts can happen in real-world (domain-class distribution sparsity), CWDG becomes an important evaluation setting for neural networks.

3.2 Proposed Method: IDFM

We propose a method to iteratively mask the features which contribute significantly to domain classification. We augment the network with an additional branch to predict the domain. The two branches are trained in parallel, one predicting the domain and another predicting the class for a given sample. In the two-branch network, the shared feature at the branched layer (the last layer) h is given by $h = f(\theta, x)$ where x is the input image and θ are the parameters of the network in all but the last layer. The class predictions and domain predictions are given by $y_{class} = f_1(\theta_1, h)$ and $y_{domain} = f_2(\theta_2, h)$. θ_1 and θ_2 are the parameters of the class and domain prediction heads, respectively.

We compute the gradient of the predicted domain with respect to h, as $grad = \partial(f_2(h;\theta_2) \cdot \hat{y}_{domain})/\partial h$, given the ground truth domain label $\hat{y}_{domain}$. Iterative Domain Feature Masking (IDFM) computes a threshold q_{th} such that the value of the top q percentile of elements in $|grad|$ is above q_{th}. q is a hyper-parameter which we choose as 33% throughout our experiments. A mask M is computed corresponding to the i^{th} element in h such that,

$$M = \begin{cases} 0 & \text{if } grad[i] \geq q_{th} \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Now we compute the masked feature h' as: $h' = h \cdot M$. After masking the dominant features corresponding to domain information, the new class prediction is $y_{class} = f_1(\theta_1, h')$. This is done iteratively in every step so that the class prediction is not reliant on domain-specific features.

The proposed IDFM method is inspired by the two previous methods, namely, Gradient reversal Ganin & Lempitsky (2015) and Representative Self Challenging (RSC) Huang et al. (2020). RSC is a single

Algorithms	PACS	VLCS	Office-home	Domain-Net	CMNIST	RMNIST	Average
IRM Arjovsky et al. (2019)	64.8	63.1	55.77	28.8	61.58	71.2	57.53
RSC Huang et al. (2020)	79.3	64.5	65.2	25.3	50.5	98.7	63.91
MMD Li et al. (2018b)	73.8	60.2	65.46	25.8	73.05	98.5	66.13
DANN Ganin et al. (2016)	74.4	64.2	62.95	24.6	72.05	98.8	66.16
MLDG Li et al. (2018a)	73	62.97	65.87	25.73	71.93	98.5	66.3
CORAL Sun & Saenko (2016)	77.06	60.2	65.46	25.8	73.5	98.5	66.75
C-DANN Li et al. (2018c)	77.7	63.77	64.58	24.04	72.5	98.9	66.91
ERM-Inc-Resnet	79.6	60.86	66.1	25.8	71.15	99.8	67.21
Mixup Xu et al. (2020)	77.6	64.2	66.02	25.1	73.05	98.3	67.35
DRO Sagawa et al. (2019)	79.38	64.77	66.1	25.15	73.05	98.5	67.82
MIRO Cha et al. (2022)	84.66	63.72	63.3	24.12	72.12	99.5	67.9
GRL Ganin & Lempitsky (2015)	86.2	65.2	66.9	26.1	74.15	99.3	69.64
IDFM (ours)	88.84	67.87	66.9	26.9	75.32	99.7	70.92

Table 1: Comparing the performance of different algorithms in DomainBedin CWDG setting. All methods are trained using the same backbone (Inception-Resnet Szegedy et al. (2017)). Algorithms are sorted by their average performance across the six datasets.

branch network that iteratively masks the features with the highest contribution towards the class prediction. The assumption here is that the dominant features activated on training data are domain-specific. We explicitly compute the features that contribute the most to domain prediction and mask them, freeing the method of the aforementioned assumption.

4 Experiments and Results

We compare the performance of eleven DG algorithms on six different datasets against the proposed IDFM method in the CWDG benchmark. This includes our implementations of ERM with Inception-ResNet Szegedy et al. (2017) backbone. We present our results on one randomly selected split of CWDG (randomly selecting a test domain for each class). The performance of ERM on a few other splits is presented in Appendix C. We use the same augmentations and hyper-parameters as in DomainBed.

Table 1 illustrates the obtained results. TDG results from the prior art are in Appendix B. A one-to-one comparison between the results in TDG and CWDG settings is not entirely meaningful. However, it is worth noting that despite seeing all domains during training, the obtained accuracies are lower than their TDG counterpart. The comparison suggests that classwise priors pose a significant challenge in OOD generalization.

Figure 1(c) shows the normalized performances of different methods on TDG and CWDG. We observe that the idea of gradient reversal (GRL) holds merit in CWDG formulation, giving notable improvements over the existing DG methods. It is worth noting that GRL deteriorated performance in TDG, further motivating the need to evaluate DG from varied perspectives. IDFM achieves state-of-the-art results, demonstrating the efficacy of the proposed approach. The results also indicate that explicit feature masking seems to improve over gradient-based feature manipulation in the studied setting, and the idea may be worth exploring in alternate settings like domain adaptation.

5 Conclusion

Performance drop of neural networks under distribution shift presents an interesting failure case. We present a new evaluation strategy beyond the Traditional Domain Generalization (TDG) called Class-wise Domain Generalization (CWDG) benchmark. In this setting, for each class, we randomly select one of the domains and keep it aside for testing. Despite being exposed to all domains during training, our experiments show that the performance of neural networks drops in this framework. Finally, we propose a novel method called the Iterative Domain Feature Masking, achieving state-of-the-art results on the proposed benchmark.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. arXiv:1907.02893*, 2019.*

Francis Bach. Breaking the curse of dimensionality with convex neural networks. The Journal of Machine Learning Research*, 18(1):629–681, 2017.*

Francesco Cappio Borlino, Antonio D'Innocente, and Tatiana Tommasi. Rethinking domain generalization baselines. In ICPR*, 2021.*

Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. arXiv preprint arXiv:2203.10789*, 2022.*

Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In ICCV*, 2013.*

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In ICML*, 2015.*

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. The journal of machine learning research*, 17(1):2096–2030, 2016.*

Xavier Gastaldi. Shake-shake regularization. arXiv:1705.07485*, 2017.*

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. Nature Machine Intelligence*, 2(11):665–673, 2020.*

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In ICCV*, 2015.*

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In International Conference on Learning Representations*, 2021. URL* `https://openreview.net/forum?id=lQdXeXDoWtI`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR*, 2016a.*

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In ECCV*, 2016b.*

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In CVPR*, 2017.*

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. ECCV*, 2020.*

Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In ECCV*, 2012.*

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. NeuIPS*, 2012.*

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In ICCV*, 2017.*

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In AAAI*, 2018a.*

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In CVPR*, 2018b.*

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In ECCV*, 2018c.*

Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In ICIP*), 2018.*

Udit Maniyar, Aniket Anand Deshmukh, Urun Dogan, Vineeth N Balasubramanian, et al. Zero shot domain generalization. arXiv:2008.07443*, 2020.*

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. Pattern recognition*, 45(1):521–530, 2012.*

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. arXiv preprint arXiv:2010.15775*, 2020.*

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In CVPR*, 2019.*

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In ICML*, 2020.*

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? arXiv:1806.00451*, 2018.*

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In ICML*, 2019.*

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv:1911.08731*, 2019.*

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556*, 2014.*

Sarath Sivaprasad, Ankur Singh, Naresh Manwani, and Vineet Gandhi. The curious case of convex neural networks. ECML-PKDD*, 2021.*

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research*, 15(1):1929–1958, 2014.*

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In ECCV*, 2016.*

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In AAAI*, 2017.*

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. arXiv:2007.00644*, 2020.*

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In CVPR*, 2017.*

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In ICML*, 2013.*

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. arXiv:2103.03097*, 2021.*

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. arXiv:1705.08292*, 2017.*

Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In AAAI*, 2020.*

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. arXiv:1611.03530*, 2016.*

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. arXiv:2103.02503*, 2021a.*

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. ICLR*, 2021b.*

A Related Work

We discuss the prior art in two components. We first address the common notion of generalization in IID data. Subsequently, we discuss the previous works on TDG and motivate the need for the new CWDG setting.

Generalization in IID setting: *Sufficiently parameterized networks can completely fit any training data Zhang et al. (2016). Hence, an essential way to evaluate a neural network is to train on a randomly selected portion of the data and test on the unseen part. Popular methods like dropout Srivastava et al. (2014), weight decay, early stopping, and regularization techniques Wan et al. (2013); Gastaldi (2017) have shown to improve this notion of generalization. It is well known that spatial transforms in image data help improve generalization. Constraining networks Bach (2017); Sivaprasad et al. (2021) have also shown to improve generalization in IID setting. The optimizer also plays a role in generalization; specifically, stochastic gradient descent is shown to achieve better generalization than adaptive algorithms Wilson et al. (2017).*

Recently Taori et al. (2020) suggest considering generalization beyond IID setting with perturbations. They report a thorough study with 204 ImageNet models, showing that robustness from synthetic image perturbations like noise, simulated weather artifacts, adversarial examples, etc., does not improve the performance on distribution shift arising in real-world data. Moreover, Recht et al. (2018, 2019) expose the problems in using a specific part of IID distribution as test data. They show a drop in performance when tested on new test data collected from the same distribution, motivating the evaluation beyond the IID setting.

Domain generalization: *Our work focuses on the DG in deep neural networks, and for pre-deep learning efforts, we refer the reader to the review by Moreno-Torres et al. (2012). Furthermore, we limit our discussion to DG in image classification. TDG formulation involves learning from multiple domains and testing on an unseen domain. TDG on image classification is commonly evaluated on six datasets: Li et al. (2017); Fang et al. (2013); Venkateswara et al. (2017); Arjovsky et al. (2019); Peng et al. (2019); Ghifary et al. (2015). The details of these datasets are illustrated in Table 2. We perform experiments on all these datasets.*

Learning domain agnostic features using the TDG formulation has seen significant interest in recent years. The problem has been approached from many different directions like data augmentation Borlino et al. (2021); Zhou et al. (2021b); Xu et al. (2020), gradient manipulation Ganin & Lempitsky (2015); Huang et al. (2020), ensemble learning Mancini et al. (2018), and feature disentanglement Khosla et al. (2012); Piratla et al. (2020). For a comprehensive list, readers can refer to the recent surveys Wang et al. (2021); Zhou et al. (2021a). It is worth noting that several of these ideas Ganin & Lempitsky (2015) have found widespread success beyond the TDG setting.

Gulrajani and Lopez-paz Gulrajani & Lopez-Paz (2021) suggest that inconsistencies in experimental conditions (datasets and training protocols) render fair comparisons difficult. They propose DomainBed, a unifying benchmark for TDG. They empirically show that a carefully implemented ERM outperforms the state-of-the-art in terms of average performance. It is reasonable to wonder why none of the numerous inventive ideas for TDG improves over the baseline ERM. In this work, we claim that TDG is not an appropriate formulation to measure the efficacy of a model to learn domain agnostic features, at least in the current form. We argue that in TDG formulation, learning domain agnostic features is most convenient for the network, not a challenge.

Consequently, we propose CWDG, a more challenging DG formulation, which leaves room for shortcut learning Geirhos et al. (2020). Our work interestingly contrasts with Maniyar et al. (2020) which proposes further constraints *on TDG by introducing unseen classes in the test domain. We instead* relax *the assumptions and expose all domains during training.*

B TDG Performance

In this section, we show that methods that help IID generalization are also key while training an ERM in TDG. We demonstrate that exploiting these IID tricks gives a competitive performance on a vanilla ERM. Following prior art in TDG, we keep aside one domain for testing in each fold and train on the other three. We use an oracle for model selection. To measure the effect of each

Dataset	# D	Domains	# C	# Images
RMNIST Ghifary et al. (2015)	6	0°, 15°, 30°, 45°, 60°, 75°	10	70000
CMNIST Arjovsky et al. (2019)	2	Red, Green	2	120000
DomainNet Peng et al. (2019)	6	Clipart, Infograph, Painting, Quickdraw, Real, Sketch	345	586575
PACS Li et al. (2017)	4	Photo, Art-Painting, Cartoon, Sketch	7	9991
VLCS Fang et al. (2013)	4	Caltech101, LabelMe, SUN09, VOC2007	5	10729
Office-Home Venkateswara et al. (2017)	4	Art, Clipart, Product, Photo	65	15558

Table 2: The table presents the statistics of datasets we use for evaluation.

	Adam with augmentation				SGD without augmentation				SGD with augmentation			
Backbones	**Photo**	**Sketch**	**Art**	**Cartoon**	**Photo**	**Sketch**	**Art**	**Cartoon**	**Photo**	**Sketch**	**Art**	**Cartoon**
Alexnet	78.05	58.72	60.56	64.13	88.26	60.42	65.645	70.065	87.69	69.17	66.91	69.28
Vgg-19_BN	80.63	69.62	69.08	70.5	88.41	77.63	76.31	70.47	84.27	82.12	70.65	79.6
Resnet-18	83.21	67.83	69.47	76.07	87.96	74.38	75.7	77.38	87.34	80.29	73.64	76.91
Resnet-50	83.08	70.85	65.31	76.97	88.53	78.21	72.99	79.28	87.57	81.02	74.91	77.72
DenseNet-121	86.21	68.72	71.64	74.81	87.5	79.1	73.4	76.44	88.9	80.42	74.31	78.15
Inc-Resnet	89.27	71.28	73.4	76.81	96.12	81.36	84.96	83.69	95.06	87.35	88.8	84.8

Table 3: The table shows DG results on PACS and the effect of various modeling choices. The compiled results compare accuracies across the two optimizing algorithms, the different backbone models, and with and without augmentation.

intervention in the ablation, we keep all the other modeling choices the same and run the experiment five times and report the mean value. Unless specifically mentioned otherwise, the training protocol and hyper-parameters are the same as in DomainBed Gulrajani & Lopez-Paz (2021). The data augmentation is also the same as in DomainBed.

The ablation experiments are limited to PACS Li et al. (2017) dataset. We train the ERM with six different backbones: AlexNet Krizhevsky et al. (2012), VGG-19 Simonyan & Zisserman (2014), ResNet-18 He et al. (2016a), ResNet-50 He et al. (2016b), DenseNet-121 Huang et al. (2017) and Inception-Resnet Szegedy et al. (2017). We run all four folds of PACS on these backbones with SGD and ADAM optimizer. As the next intervention, we run all the aforementioned backbones with and without augmentations with an SGD optimizer.

We compare the optimized ERM baseline against the top-performing methods on DomainBed, on six different datasets (Table 2). Outside of DomainBed, we also use the multi-branch reverse-gradient (GRL) Ganin & Lempitsky (2015) model on the Inception-ResNet backbone.

B.1 Exploring effects of different modeling choices

Effect of optimizer: *Table 3 compiles the accuracy of different backbones under SGD and ADAM over all domains in PACS. SGD outperforms ADAM across all the backbones. Also, the performance of ADAM is highly susceptible to the learning rate. For instance, averaged over the four domains, SGD gives 89.00% accuracy compared to 77.69% accuracy given by ADAM, using the Inception-Resnet backbone. With a higher learning rate (same as SGD), ADAM gives only 48.44%. The results show that SGD has a clear advantage over ADAM in the studied scenario. The observation may stem from the fact that fine-tuning a large ImageNet model on a relatively small dataset like PACS is an 'overparameterized problem'. Wilson et al. (2017) suggests that for simple overparameterized problems, adaptive methods can find drastically different solutions than SGD.*

Effect of augmentation: *Table 3 compares the accuracy of different backbones trained using SGD, with and without augmentation. We observe that augmentations almost always improve the performance of networks. For instance, with the Inception-Resnet model, the average performance of the model across all four domains with augmentation is 89.00%, which is higher than the average accuracy without augmentation 86.53%.*

Effect of choice of backbone: *Table 3 shows the significance of backbone in DG. Across all domains and irrespective of augmentation and other choices, the Inception-Resnet backbone outperforms all other backbones. Zhou et al. (2021a) questions the common perception that models that perform on ImageNet will learn domain-generalizable features and hence argues for DG tailored*

Algorithms	PACS	VLCS	Office-home	Domain-Net	CMNIST	RMNIST	Average
IRM Arjovsky et al. (2019)	82.9	77.2	66.7	32.6	59.16	97.7	69.37
GRL Ganin & Lempitsky (2015)	83.69	77.38	70.2	37.4	50.5	98.49	69.61
MMD Li et al. (2018b)	82.8	76.7	67.1	28.4	73.35	98.1	71.07
DANN Ganin et al. (2016)	84	77.7	65.5	38.1	73.03	89.1	71.23
C-DANN Li et al. (2018c)	81.7	74	64.7	37.9	73.03	96.3	71.27
DRO Sagawa et al. (2019)	83.1	77.5	67.1	33.4	73.35	97.9	72.05
RSC Huang et al. (2020)	84.77	78.8	70.8	39.2	61.2	98.23	72.16
MLDG Li et al. (2018a)	82.4	77.1	67.6	41.6	71.64	98	73.05
Mixup Xu et al. (2020)	83.7	78.6	68.2	38.7	73.34	98.1	73.44
CORAL Sun & Saenko (2016)	83.6	77	68.6	40.2	73.35	98.1	73.47
ERM-Inc-Resnet	89.11	78.84	71.95	43.2	74.35	99.2	76.10

Table 4: Comparing ERM-Inc-Resnet with other algorithms in DomainBed. The algorithms are sorted by their average performance across the six datasets.

Classes	Domain	Domain	Domain	Domain	Domain
Guitar	Photo	Art	Cartoon	Sketch	Art
Person	Photo	Cartoon	Art	Photo	Cartoon
Horse	Cartoon	Sketch	Cartoon	Art	Photo
Elephant	Sketch	Photo	Art	Sketch	Art
Dog	Photo	Art	Sketch	Cartoon	Cartoon
Giraffe	Art	Photo	Cartoon	Art	Photo
House	Cartoon	Cartoon	Photo	Sketch	Sketch
Accuracy	79.6	79.38	78.81	79.12	79.18

Table 5: Performance of NN on different train test splits in CWDG setting. The last row of each column shows the results of one run, and each row of the column corresponds to the domain kept out for the corresponding class.

methods. In contrast, we observe that the better performing backbone for DG is the better performing model on the ImageNet IID benchmark and not necessarily the backbone with more parameters.

B.2 Comparing with baselines in TDG

In Table 4, we compare the ERM baseline against other algorithms in DomainBed Gulrajani & Lopez-Paz (2021). The proposed ERM baseline with Inception-Resnet backbone (ERM-Inc-Resnet) not only outperforms other methods on average but also outperforms the best performing model in every dataset. On the PACS dataset, we get a margin of above 5% from the next best performing model. This comparison shows that neural networks trained with a robust backbone, augmentation, and optimizer under TDG setting do not need any additional method to learn domain agnostic features. We find that all the studied methods fail to give any improvements over the ERM baseline. In fact, some of the methods simply inhibit learning, reducing test performance. This motivates us to think if, by solving deficits of neural networks in TDG setting, are we attempting to fix a system that is not broken?

C Different seeds for CWDG

In this section, we present our results for the different train test splits in CWDG setting of the PACS dataset. We observe comparable performance across the different seeds of CWDG (Table 5). Each column in the table correspond to one run, and the last row in each column shows the accuracy of ERM Inc-ResNet on that run. That is, each row corresponds to a class, and each element shows the domain kept out for the particular class. The first column shows the split used in the main text. The accuracies show that as long as the domains are evenly spread such that there is a clear prior in the train split, the performance of the neural network stays significantly below the TDG setting.